DIME DYNAMIC DOCUMENTATION TRAINING

Luiza Andrade & Mrijan Rimal

June 8, 2017

Contents

Introduction

This exercise is an introduction to the concepts needed to create a docuement with tables and figures outputted in LaTeX from, for example, Stata or R. After this exercise you will be able to create a document that is automatically updated each time the code in Stata or R is updated and run again. The document this exercise generates is meant to be either an appendix of tables and graphs, or a document with initial results that updates instantly.

For this first exercise, all the outputed tables and graphs that you need to create the document are provided for you. In later exercises you will produce them yourself.

If you want to have a more broad understanding of LaTeX or explore other functionalities, this LaTeX Wikibook is a great source. For more specific questions about how to perform a task or solve an error, Google usually has the answer. However, the exercises following this one will introduce you to more functionalities in LaTeX.

Exercise overview:

- Part 1 and 2 explains concepts often confusing to someone new to LaTeX.

- Part 3 to 6 imports figures and tables and introduces some basic formatting.

- Part 7 and 8 introduces document titles and list of tables and figures.

1 Folders and relative file paths

LaTeX uses what is called relative file paths. We will explain what this means and while it is not complex, it is important to understand it well when working in LaTeX. Start by looking in the folder where you found this handout.

In this folder, you see a folder called *Output.tex* with two sub-folders, *Raw* and *Final*. In the *Final* sub-folder, you will find a file called *Exercise 1.tex*. This is the file you will be working in. We will call it the main document and you will import the tables and graphs we have provided for you to this main document file. You will also find a folder called *Raw*. In this folder you will find the tables and graphs you will use in this exercise and that we have provided for you. The relative file path from *Exercise 1.tex* to, for example, image *iegraph.png* in the *Raw* folder is therefore {../Raw/iegraph.png}. Note that to a relative file path it does

not matter where the file is stored on the computer(for example, on C drive, My Documents, the Drop-Box folder, Desktop etc). It only matters where a file is saved in relation to the starting point. The starting point in LaTeX is the folder where the main document is saved. In the case of this exercise that is the folder where *Exercise 1.tex* is saved.

So this path `{../Raw/iegraph.png}` means that, starting to the *Final* folder where our main document is, first LaTeX will go its parent folder (that is what the double dots do), and then look there for another folder called *Raw*, and from that folder open a file called *iegraph.png*. The file path `{iegraph.png}` on the other hand means that the graph would be saved directly in the same folder as the main document. This is usually bad practice as when your document grows it will reference a lot of files and storing all those files in a sub-folder is much cleaner way to organize the files. Relative file paths can be multiple folders deep, for example `{../Raw/baseline/income_chapter/iegraph.png}`.

Relative file paths are very simple as long as the files referenced are stored in close relation to the main document, but this is soon obvious to you unless it already is.

2   Compiling a pdf file from LaTeX code

In LaTeX, compiling means that you run your code to create and save a pdf version of your document. It is similar to running a do-file in Stata or an R-script in R but it creates a pdf instead of running computations.

In TeXstudio, open the file *Exercise 1.tex* that you find in the folder for this exercise. This is a very basic LaTeX document and consists on the initial settings we will build on. The preamble at the top of the document loads different functionalities that we will be using, but you do not have to worry about this until later exercises.

To compile the document go to *Tools* in the top drop-down menu in TeXstudio when you have the code in *Exercise 1.tex* open. Then click *Compile*, but you wont to see anything happening. Instead, click *Build & View* in the *Tools* menu. This should open a new window in TeXstudio where you see the text *This is a blank document*. Both *Compile* and *Build & View* creates a pdf and saves it to disk, but only *Build & View* opens up a window where you can see your document directly.

The pdf file was saved in the same folder as where the file *Exercise 1.tex* is saved. LaTeX exports a lot of auxiliary files as well here. You do not need these files. It is possible to have these files saved elsewhere, but

do not worry about that now, and most LaTeX users never bother to change where those files are saved.

If you open the *Tools* menu again you can see short cuts for next time you want to compile your document. You can click *F5* to *Build & View* and *F6* to *Compile*. You also see two icons that look like green play-buttons. You can find the same icons in the bar underneath the drop-down menus. Most LaTeX users use either of those two short cuts.

3   Using TeXstudio interface to import an image into LaTeX

Now it is time to start importing some files into our document. Make sure that you still have *Exercise 1.tex* open. Again, do not worry about the preamble, for the rest of this exercise, you will only add lines of code between \begin{document} and end{document} to create your document content.

We first want to show you a short cut that you can use to import files to your document that you can use until you feel confident in using relative file paths. While you keep *Exercise 1.tex* open in TeXstudio, open the *Raw* folder in regular Explorer (Windows) or Finder (Mac) window. Now click *take_up.png* and drag it to the part of *Exercise 1.tex* that says "This is a blank document". Then click "OK" in the window that pops up. This will create a section of code that should look similar to the code below. Now, compile your document using *Build & View*.

```
\begin{figure}
        \centering
        \includegraphics[width=0.7\linewidth]{../Raw/take_up}
        \caption{}
        \label{fig:take_up}
\end{figure}
```

Note that the relative file path to your *Raw* folder was created in end of this line:

```
\includegraphics[width=0.7\linewidth]{../Raw/take_up}
```

In this example, the relative file path is {../Raw/take_up} Note that the file extension is needed. You will see \begin{} and \end{} a lot in LaTeX. The define where one object start and end, and the lines of code in between \begin{} and \end{} is called an environment. In the example above it is a figure environment. This tells LaTeX that you are importing a figure and any other code in that environment in addition to the

line with the relative path to the figure is a setting or provide some meta information to that figure. You will learn what `\centering` and `\caption{}` below. `\label{fig:take_up}` is a slightly more advanced option and we will explore it on *Exercise 2*.

4   Manually importing graphs into LaTeX

While click-and-drag is helpful to get started, you need to learn how to import figures and tables manually to LaTeX. So now we want to manually import a second figure to your document. This will help us understand the functions of the other lines of code that TeXstudio created for us in the click-and-drag example.

Task 1: Below the code that was generated when you imported the first figure (but before `\end{document}`), create a figure environment and import *iegraph.png* into it. You will find that graph in the *Raw* folder. You import that figure by typing the following lines of code:

```
\begin{figure}[H]
        \includegraphics{../Raw/iegraph.png}
\end{figure}
```

Remember to include `../Raw/` as this is the relative path to the *Raw* folder where *regular_graph.png* is saved. Compile your document. How does it look?

Task 2: We need to adjust the size of your figure so it fits into the page. Use the `\width` option of `\includegraphics{}` to do this by typing the context in blue:

```
\begin{figure}[H]
        \includegraphics[width=\textwidth]{../Raw/iegraph.png}
\end{figure}
```

The property `=\textwidth` means that the figure will be as wide as the margins of the text in your document. There are more properties that can be used with `\width`, and you will learn more about these in later exercises, but for now, you can keep using only `=\textwidth` for all figures and tables where it is necessary.

Task 3: Add a title to your figure using the `\caption` command:

```
\begin{figure}[H]
        \includegraphics[width=\textwidth]{../Raw/iegraph.png}
        \caption{Add figure title here}
\end{figure}
```

Change the caption from `{Add figure title here}` to any title of your choice and compile the document again.

Part 5. Importing tables into LaTeX

Task 1: Create a table environment by using `\begin{}` and `\end{}` underneath the two figures you already imported. Instead of writing `figure` in `\begin{}` and `\end{}` as you did above, write `table`. This tells LaTeX to expect a table instead if a figure. Import *regression_table.tex* into the table environment using `\input{}`. This can be done by typing the following lines:

```
\begin{table}[H]
        \input{../Raw/regression_table.tex}
\end{table}
```

Compile the document and look at the table.

Task 2: We need to adjust the size of the table. When we are using `\input{}`, we have to do that slightly differently to how we did it for *iegraph.png* above. Use `\adjustBox` to adjust table size. Everything added between `\begin{adjustbox}` and `\end{adjustbox}` will have the property `{max width=\textwidth}`. See the code in blue below:

```
\begin{table}[H]
        \begin{adjustbox}{max width=\textwidth}
        \input{../Raw/regression_table.tex}
        \end{adjustbox}
\end{table}
```

Task 3: Similar to figures, you can add a caption to your table. This is done by adding the same lines as for figures.

```
\begin{table}[H]
        \caption{Add a title to this table}
        \begin{adjustbox}{max width=\textwidth}
                \input{../Raw/regression_table.tex}
        \end{adjustbox}
\end{table}
```

Update the text in `\caption{Add figure title here}` and compile the document. See that the title of the table is above the table. When we used `\caption{}` for the figure earlier the title was below the table. That has nothing to do with tables and figures, it all depends if you enter the line with `\caption{}` before or after the line when you are importing a file to your document. To test this, move the line with the caption to after `\end{adjustbox}` and before `\end{table}`, and compile the document again. Note that the caption line must come before the adjustbox environment begins, otherwise the document will not compile.

5 Centering imported objects

Until now, we imported object into our document that were too wide for our document margins, so we had to adjust their sizes. Next, we are going to see what happens when you import a narrower object. To do this, create a table environment, import and create a title for the table in *sampl_sizes.tex*. This table can also be found in the *Raw* folder:

```
\begin{table}[H]
        \caption{Add a title to this table}
        \input{../Raw/sample_sizes.tex}
\end{table}
```

As you can see, this table is not yet centered, but we can fix that by adding one line to its environment:

```
\begin{table}[H]
        \centering
        \caption{Add a title to this table}
        \input{../Raw/sample_sizes.tex}
\end{table}
```

Adding `\centering` to the beginning of an environment centers whatever is in that environment. That

means that if the same line was added after a `\begin{figure}`, that figure would also be centered.

6   Adding a document title

We are almost done with this document now but so far we only have tables and figures. In this part we want to add a title to the document and in the next part we will add a list of tables and a list of figures. All of these are very easy to set up.

To add a title to your document, add `\maketitle` on a new line immediately below `\begin{document}`. Compile the document. You can see the information we already added. Update this information with a title of your own and with you name in the preamble. Look for this section and update the text that is blue below.

```
% ADD YOUR PROJECT INFO HERE
\title{DIME template for \LaTeX \\ Importing tables}    % Double backslash starts a new line
\author{Luiza Andrade \& Mrijan Rimal}
\date{}                                                 % Updates and prints date automatically
```

7   Add a list of tables and a list of figures

Finally we want to create a list of tables and figures that is automatically updated if you were to add or remove tables and figures. Immediately after where you added `\maketitle`, on a new line, add `\listoftables` and, on a second new line, add `\listoffigures`. Like this:

```
\begin{document}
\maketitle
\listoftables
\listoffigures
```

Compile your document. The titles shown in these lists are those defined by the `\caption` you added to each table and figure. Go to any of your figures or tables and change the caption and compile the document again. See how the list of tables and figures is updated. While it is common practice, you do not need to add `\maketitle` and `\listoftables` in the beginning of the document. You can add them anywhere.